202-1092

METHOD AND SYSTEM FOR CORRECTING SENSOR OFFSETS

Related Applications

[0001] The present invention claims priority to provisional application no. 60/401,417 filed on August 5, 2002.

Technical Field

[0002] The present invention relates generally to a dynamic behavior control apparatus for an automotive vehicle, and more specifically, to a method and apparatus for correcting offsets in vehicle dynamic sensors.

Background

[0003] Dynamic control systems for automotive vehicles have recently begun to be offered on various products. Typical dynamic control systems include sensors corresponding to various vehicle dynamics. Examples thereof include: a yaw rate sensor, a roll rate sensor, a longitudinal acceleration sensor, and a lateral acceleration sensor. Typical dynamic control systems also include a controller that receives the sensor signals and controls various safety and stability systems in response thereto.

[0004] During various different phases of vehicle operation, the aforementioned sensors tend to generate errors that may result in false signals received in the controller.

[0005] It would therefore be desirable to provide a correction or compensation system and method. The new system should not require additional sensors and should also not require shutting-down dynamic control sensors. The present invention is directed to these ends.

Summary of the Invention

[0006] It is therefore an object of the invention to provide a system for correcting for offsets within vehicle dynamic sensors.

[0007] In one aspect of the invention, a sensor offset correction method for a vehicle includes generating a first offset correction signal for a vehicle dynamic sensor at a sensor power-up, generating a second offset correction signal for the vehicle dynamic sensor when the vehicle is moving, and correcting the vehicle dynamic sensor in response to the first offset correction signal and the second offset correction signal.

[0008] In a further aspect of the invention, a control system for an automotive vehicle having a vehicle body includes a cluster of vehicle dynamic sensors positioned within the vehicle body adapted to generate a plurality of vehicle dynamic signals.

[0009] The system may further include a controller adapted to receive the plurality of vehicle dynamic signals, generate a first offset correction signal for one of the cluster of the vehicle dynamic sensors in response to a DC bias and at a sensor power-up, generate a second offset correction signal for the one

of the cluster of the vehicle dynamic sensors in response to a signal equivalent to a temperature drift signal and when the vehicle is moving, generate a third offset correction signal for the one of the cluster of the vehicle dynamic sensors when the vehicle is at rest and the one of the cluster of the vehicle dynamic sensors is below an accuracy threshold, and correct the one of the cluster of the vehicle dynamic sensors in response to the first offset correction signal, the second offset correction signal and the third offset correction signal.

[0010] One advantage of the invention is that readings from vehicle dynamic sensors are more accurate.

[0011] Other objects and features of the present invention will become apparent when viewed in light of the detailed description of the preferred embodiment when taken in conjunction with the attached drawings and appended claims.

Brief Description Of The Drawings

[0012] Figure 1 is a diagrammatic view of a vehicle with variable vectors and coordinator frames in accordance with one embodiment of the present invention.

[0013] Figure 2 is a block diagram of the vehicle sensor system from Figure 1.

[0014] Figure 3 is a logic flow diagram of a method for compensating for offset errors in vehicle dynamic

sensors in accordance with another embodiment of the present invention.

Detailed Description Of The Present Invention

[0015] In the following figures the same reference numerals are used to identify the same components.

[0016] Referring to Figures 1 and 2, a control system 18 for an automotive vehicle 19 having a sensing system 16 (sensor cluster) and a controller 26, is illustrated. Various forces and moments are acting thereon during a rollover condition.

[0017] The vehicle control system 18 includes the sensor system 16. The sensing system 16 may use a six control sensor set including three axial accelerometers including a lateral accelerometer 27 (generating a lateral acceleration signal), a longitudinal accelerometer 28 (generating a longitudinal acceleration signal), and a vertical accelerometer 29 (generating a vertical acceleration signal) and three axial rotation rate detectors including a yaw rate sensor 30 (generating a yaw rate signal), a roll rate sensor 31 (generating a roll rate signal), and a pitch rate sensor 32 (generating a pitch signal). Of course, those skilled in the art will recognize that one or more sensors may not be included in the system 18.

[0018] The sensor system 16 further includes various other sensors, such as wheel speed sensors 20, a steering angle sensor 33 (hand-wheel sensor), and steering angle position sensors 34 (road-wheel sensors). As one skilled in the art will understand,

the various sensors generate a plurality of vehicle dynamic signals. These are further described below.

[0019] The vehicle control system 18 may also include the controller 26. The controller 26 receives the plurality of vehicle dynamic signals and generates a first offset correction signal for one of the cluster 16 of the vehicle dynamic sensors (27, 28, 30, or 31) in response to a DC bias and at a sensor power-up. The controller 26 also generates a second offset correction signal for the one of the cluster 16 of the vehicle dynamic sensors (27, 28, 30, or 31) in response to a signal equivalent to a temperature drift signal and when the vehicle 19 is moving. The controller 26 further generates a third offset correction signal for the one of the cluster 16 of the vehicle dynamic sensors (27, 28, 30, or 31) when the vehicle 19 is at rest and the sensor is below an accuracy threshold. The controller 26 still further corrects the vehicle dynamic sensing system 16 in response to the first offset correction signal, the second offset correction signal and the third offset correction signal.

[0020] Based upon inputs from the sensor system 16, the controller 26 may control a safety device 38. Depending on the desired sensitivity of the system and various other factors, not all the sensors are used in a commercial embodiment. The safety device 38 may control an airbag 40 or a steering actuator or braking actuator at one or more of the wheels of the vehicle 19. Also, other vehicle components such as a suspension control 48 are used to adjust the

suspension to prevent rollover. Suspension control 48 may include an anti-roll bar.

[0021] In this system 18, the output signals from the RSC sensors (Yaw Rate Sensor 30, Roll Rate Sensor 31, Longitudinal Acceleration Sensor 28, Lateral Acceleration Sensor 27) are corrected for errors by removing the zero output DC bias. Such bias constitutes an error that may occur as a result of temperature changes, manufacturing defects, or other factors. This system 18 also compensates for the drift in the sensor output signals that can occur during vehicle operation.

[0022] In one embodiment, to reduce run time, calculations of the offset variables are not performed during Anti-Lock Brake (ABS), Active Yaw Control (AYC), Traction Control System (TCS), or Roll Stability Control (RSC) events. Runtime for the system 18 is highest during these events. The signals, however, are filtered and compensated during this time. Calculations of offset variables are also paused when the flag is set (one example of logic therefore includes: RSC_SENSORS_DISTURBED). This flag is set externally when the sensors are undergoing a self-test, the sensor values are too high, or the rate of change is higher than expected.

[0023] An example of logic therefore includes:

If (ABS_CYCLE || AYC_CYCLE || RSC_CYCLE || TCS_CYCLE); (In generic terms, if ABS, AYC, TCS or RSC are active then the following logic runs.)

{offset_comp()}; (This function compensates and filters out the signals from sensor.)

else{if(!SENSORS_DISTURBED){offset_initializaton(); (This function initializes the offset values.)

if (VEHICLE_STANDSTILL) {offset_resting(); (When the vehicle 19 is standing still, this function is called.)

else {offset_signals();} (When the vehicle 19 is moving, this function is called.)

{offset_comp()} (This function compensates and filters out the signals from sensors.)

[0024] One example of offset compensation external inputs to the aforementioned logic includes:

RSC Code Name	Logic Description Name
1. Get_abs_cycle	ABS_CYCLE
2. Get_ayc_in_cycle	AYC_CYCLE
3. st_ROLL_FLAG.bf_bool_ACTIVE_ROLL_CONTROL	RSC_CYCLE
4. Get_tcs_engine_control_active	TCS_CYCLE
5. RSC_SENSOR_BITS.RSC_SENSORS_DISTURBED	RSC_SENSORS_DISTURBED
6. ss_mps_REF_VELOCITY_YC	VEHICLE_SPEED
7. Get_vehicle_standstill	VEHICLE_STANDSTILL

[0025] There are three phases of operation occurring within the controller 26: 1) Initialization, normally occurring at electric power-on of the vehicle 19; 2) Resting, occurring whenever the vehicle 19 is not in motion; and 3) Dynamic, occurring when the vehicle 19 is in motion.

[0026] Referring to Figure 3, a logic flow diagram 60 of a method for compensating for offset errors in vehicle dynamic sensors (27, 28, 30, or 31) is

illustrated. Logic starts in operation block 62 where a first offset correction signal or initialization for a vehicle dynamic sensor (27, 28, 30, or 31) is generated at a sensor power-up.

[0027] Initialization of Roll Rate or roll rate signal and Yaw Rate or yaw rate signal occurs when the vehicle ignition key is turned on or electrical power is otherwise supplied to the vehicle systems. INITIALIZATION_COMPLETE indicates a flag is set at the end of initialization. Initialization eliminates initial D.C. bias that is present at the initialization.

[0028] If the vehicle 19 is standing still when the ignition is turned on, the offsets are computed such that the filtered roll rate and yaw rate is approximately zero deg/sec. Other vehicle movement checks when standing still, such as rolling or turning, are also performed. Initialization continues until the initialization timer reaches, for example, one second. If the vehicle 19 starts moving before initialization is complete, the average of the maximum and minimum offset values (that have been calculated over a long time period, such as the entire operating time of the vehicle 19) are computed and used as the initial offset values.

[0029] In one embodiment, the average of the yaw rate and roll rate is limited to between, for example, +-3.5deg/sec, because the sensor specification defines the +-3.5deg/sec example as the worst case zero point offset for the rate sensors.

[0030] Initialization occurs when the vehicle ignition is turned on, or electrical power is otherwise supplied to the vehicle systems. If the vehicle 19 is standing still when the ignition is turned on, then the offsets are computed such that the filtered Ax (Longitudinal Acceleration) and Ay (Lateral Acceleration) values are approximately equal to the Ax and Ay values that existed and were written to the controller EEPROM 47(Electrically Erasable Programmable Read-Only Memory) when power was removed from the system 18 at the end of the previous driving cycle.

[0031] If the vehicle 19 was not standing still when the previous EEPROM 47 write occurred or an EEPROM 47 write did not occur during the last ignition off, then the offsets is computed by taking the average of the maximum and minimum value. The system assumes that the geographical location of the vehicle 19 has not been changed since the vehicle 19 has been shut off. Also to minimize error if the vehicle 19 has been moved during an ignition off, the Ax and Ay offset calculations are limited to within the boundaries of maximum and minimum values from EEPROM 47.

[0032] An example of initialization logic includes:

```
If ((VehicleStandStill)&& !(INIT_FLAG & INITIALIZATION_COMPLETE){if ((RollAcceleration <= VEHICLE_STAND_STILL_ROLL_ACC)&& (YawAcceleration <= VEHICLE_STAND_STILL_YAW_ACC));
```

```
{if (!(RSC_SENSORS_DISTURBED))
    {RollRateOffset += FilteredRollRate / InitializationTimer;
    YawRateOffset += FilteredYawRate / InitializationTimer;
    InitializationTimer ++;}
    If (InitializationTimer >= ONE_SEC_UP)
    {INIT_FLAG |= INITIALIZATION_COMPLETE;}}
    Else
    {InitializationTimer = 0;     /* Restart the initialization */}}
    else
    {RollRateOffset = LIMIT(+-3.5deg/sec, (MAX_RR_OFST_EE + MIN_RR_OFST_EE )/2);
    YawRateOffset = LIMIT(+-3.5deg/sec, (MAX_YR_OFST_EE + MIN_YR_OFST_EE)/2);
    INIT_FLAG |= INITIALIZATION_COMPLETE;}
```

[0033] For the aforementioned logic, one example of offset initialization external inputs includes:

RSC Code Name	Logic Description Name
8. Get_vehicle_standstill(): A macro to determine if the vehicle is standing still	VehicleStandStill
9. ss_mpss_LONG_ACC: Raw Longitudinal Acceleration	RawAx
10. ss_mpss_LAT_ACC: Raw Lateral Acceleration	RawAy
11. ss_dps_FLT_ROLL_RATE: Filtered and compensated roll rate.	FilteredRollRate
12. ss_dps_FLT_YAW_RATE: Filtered and compensated yaw rate.	FilteredYawRate
13. ROLL_ACCELERATION2: Roll Rate Acceleration.	RollAcceleration
14. YAW_ACCELERATION2: Yaw Rate Acceleration.	YawAcceleration

15. RSC_SENSOR_CALIBRATION_VALUES [Rr_ofst_max]: Roll rate maximum value from EEPROM (over a long time period, such as total operating time of the vehicle)	MAX_RR_OFFSET_EE
16. RSC_SENSOR_CALIBRATION_VALUES [Rr_ofst_min]: Roll rate minimum value from EEPROM (over a long time period, such as total operating time of the vehicle)	MIN_RR_OFFSET_EE
17. RSC_SENSOR_CALIBRATION_VALUES [Yr_ofst_max]: Yaw rate maximum value from EEPROM (over a long time period, such as total operating time of the vehicle)	MAX_YR_OFFSET_EE
18. RSC_SENSOR_CALIBRATION_VALUES [Yr_ofst_min]: Yaw rate minimum value from EEPROM (over a long time period, such as total operating time of the vehicle)	MIN_YR_OFFSET_EE
19. RSC_SENSOR_CALIBRATION_VALUES [Along_ofst_max]: Longitudinal Acceleration maximum value from EEPROM (over a long time period, such as total operating time of the vehicle)	MAX_AX_OFFSET_EE
20. RSC_SENSOR_CALIBRATION_VALUES [Along_ofst_min]: Longitudinal Acceleration minimum value from EEPROM (over a long time period, such as total operating time of the vehicle)	MIN_AX_OFFSET_EE
21. RSC_SENSOR_CALIBRATION_VALUES [Alat_ofst_max]: Lateral Acceleration maximum value from EEPROM (over a long time period, such as total operating time of the vehicle)	MAX_AY_OFFSET_EE
22. RSC_SENSOR_CALIBRATION_VALUES [Alat_ofst_min]: Lateral Acceleration minimum value from EEPROM (over a long time period, such as total operating time of the vehicle)	MIN_AY_OFFSET_EE
23. RSC_SENSOR_CALIBRATION_VALUES [Long_acc] : Longitudinal Acceleration stored in EEPROM from the previous ignition cycle	AxEE
24. RSC_SENSOR_CALIBRATION_VALUES [Lat_acc]: Lateral Acceleration stored in EEPROM from the previous ignition cycle	AyEE

25. Get_rsc_eeprom_var_bit(VEHICLE _STAND_STILL): A macro to determine from EEPROM if vehicle was standing still last time the EEPROM write was made.	Vehicle_was_standing_still_during_last_ignition_off
26. Get_rsc_eeprom_var_bit(NO_EEPROM _STORAGE_RSC): A macro to determine from EEPROM if an EEPROM write is made in the last ignition cycle.	An_EEPROM_write_was_made_during_last_ignition_off

[0034] One example of offset initialization outputs include:

RSC Code Name	**Logic Description Name**
1. ss_mpss_LAT_ACC_OFFSET	AxOffset
2. ss_mpss_LONG_ACC_OFFSET	AyOffset
3. ss_mpss_ROLL_RATE_OFFSET	RollRateOffset
4. ss_mpss_YAW_RATE_OFFSET	YawRateOffset
5. ss_mpss_ROLL_RATE_OFFSET_INIT: The initial value of roll rate offset.	InitialRollRateOffset
6. ss_mpss_YAW_RATE_INIT: The initial value of filtered and compensated yaw rate.	InitialYawRate
7. Clr_rsc_eeprom_var_bit(NO_EEPROM _STORAGE_LAST_CYCLE): A macro that clears the specified bit in shadow EEPROM.	
8. bf_uc_INITIALIZATION_STATUS: Flags set during initialization	INITIALIZATION _COMPLETE

[0035] Offset initialization parameters include: rr_ofst_eep_max, which is the maximum limit that the roll rate offset can be; yr_ofst_eep_max, which is the maximum limit that the yaw rate offset can be; along_ofst_eep_max, which is the maximum limit that the longitudinal acceleration offset can be; alat_ofst_eep_max, which is the maximum limit that the lateral acceleration offset can be.

[0036] One example of offset initialization state variables includes: bf_uc_initialization_status, which is an initialization flag for checking initialization condition status, and ss_tim_initialization_timer, which sets a start time for the initialization timer.

[0037] In one embodiment of the present invention, if the vehicle 19 starts moving before the initialization completes, a flag is set, and an initial value of the Ax and Ay offsets is computed by taking the maximum and minimum offsets values from EEPROM 47. When the initial value is calculated by taking the average of the maximum and the minimum value, it is always limited to, for example, +-1.5m/sec^2 for the acceleration sensors, because the acceleration sensors should not experience the worst case zero point offset of greater than, for example, 0.11g. An example of logic therefore includes:

```
If ((VehicleStandStill)
&&!(INIT_FLAG & INITIALIZATION_COMPLETE)
&&(Vehicle_was_standing_still_during_last_ignition_off)&&(An_EEPROM_write_was_made_during_last_ignition_off))
{if((RollAcceleration<=VEHICLE_STAND_STILL_ROLL_ACC)
&& (YawAcceleration <= VEHICLE_STAND_STILL_YAW_ACC))
{if (!(RSC_SENSORS_DISTURBED))
{AxOffset += (RawAx - AxEE) * LOOP_TIME_SEC;
AyOffset += (RawAy - AyEE) * LOOP_TIME_SEC;
AxOffset = max (MIN_AX_OFST_EE, AxOffset);
AxOffset = min (MAX_AX_OFST_EE, AxOffset);
```

AyOffset = max (MIN_AY_OFST_EE, AyOffset);

AyOffset = min (MAX_AY_OFST_EE, AyOffset);

InitializationTimer ++;}

If (InitializationTimer >= ONE_SEC_UP) {INIT_FLAG |= INITIALIZATION_COMPLETE;}}; an initial value of the Ax and Ay offsets is computed by taking the maximum and minimum offsets values from EEPROM 47.

else

{InitializationTimer = 0; /* Restart the initialization */}}

else{AxOffset = LIMIT(+-1.5m/sec^2,(MAX_AX_OFST_EE + MIN_AX_OFST_EE)/2); limits the Ax offset max and min to 1.5m/sec^2.

AyOffset = LIMIT(+-1.5m/sec^2, (MAX_AY_OFST_EE + MIN_AY_OFST_EE)/2); limits the Ay offset max and min to 1.5m/sec^2.

INIT_FLAG |= INITIALIZATION_COMPLETE;}

[0038] In operation block 64, a second offset correction signal is generated for the vehicle dynamic sensor (27, 28, 30, or 31) when the vehicle 19 is moving. The second offset signal deals with moving conditions of the vehicle. Embodiments thereof are discussed below.

[0039] If the vehicle 19 is turning continuously for a given time period (15 seconds is used in this example) in one direction, offset compensation may be paused for yaw rate, roll rate, and Ay. If the vehicle 19 is turning continuously to the right hand side, a negative turn flag is set and if it is turning to the left hand side, a positive turning flag is set. Logic to illustrate this includes:

```
If ((FltYawRate < 0 && SWA < RIGHT_TURN)
{POS_TURNING_COUNTER -= 2;
NEG_TURNING_COUNTER ++;
POS_TURNING_COUTNER = min (POS_TURNING_COUNTER,
MAX_COUNT_VALUE);
NEG_TURNING_COUNTER = max (NEG_TURNING_COUNTER,
MIN_COUNT_VALUE);
If (TURNING_TIME < NEG_TURNING_COUNTER)
SET_SUSTAINED_NEG_TURNING_15SEC;
Else
{If (POS_TURNING_COUNTER <= TURNING_TIME)
CLEAR_SUSTAINED_POS_TURNING_15SEC;
CLEAR_SUSTAINED_NEG_TURNING_15SEC;}}
If ((FltYawRate > 0 && SWA > LEFT_TURN)
{NEG_TURNING_COUNTER -= 2;
POS_TURNING_COUNTER ++;
NEG_TURNING_COUTNER = min (NEG_TURNING_COUNTER,
MAX_COUNT_VALUE);
POS_TURNING_COUNTER = max (POS_TURNING_COUNTER,
MIN_COUNT_VALUE);
If (TURNING_TIME < POS_TURNING_COUNTER)
SET_SUSTAINED_POS_TURNING_15SEC;
Else{If(NEG_TURNING_COUNTER<= TURNING_TIME)
CLEAR_SUSTAINED_NEG_TURNING_15SEC;
CLEAR_SUSTAINED_POS_TURNING_15SEC;}}
```

[0040] To clarify the aforementioned logic, an example of dynamic offset external inputs includes:

[0041] An example of dynamic offset variables includes:

RSC Code Name	**Logic Description Name**
1. Get_vehicle_standstill()	
2. bf_uc_INITIALIZATION_STATUS	
3. ss_deg_STEERING_WHEEL_ANGLE	
4. Get_ayc_reverse_det_possible()	
5. ss_dps_FLT_YAW_RATE	FltYawRate
6. ss_deg_PITCH_ANGLE_ESTIMATE	PitchAngleEst
7. ss_deg_PITCH_OFFSET_ESTIMATE	PitchOffset
8. ss_deg_REL_PITCH_ANGLE	PitchRelative

[0042] An example of dynamic offset outputs includes:

RSC Code Name	**Logic Description Name**
1. ss_dps_ROLL_RATE_OFFSET	RollRateOffset
2. ss_dps_YAW_RATE_OFFSET	YawRateOffset
3. ss_dps_LONG_ACC_OFFSET	AxOffset
4. ss_dps_LAT_ACC_OFFSET	AyOffset

[0043] One embodiment of dynamic offset parameters includes: TURNING_TIME; MAX_COUNT_VALUE; MIN_COUNT_VALUE; Rr_ofst_eep_max; Yr_ofst_eep_max; Alat_ofst_eep_max; Along_ofst_eep_max; and p_LOOP_TIME_SEC.

[0044] Under most normal driving circumstances, the average yaw rate and the average roll rate should be close to zero over a long period of time. Any time a non-zero value is detected for the yaw rate and/or roll rate, they are compensated for continuously by a very small amount, which is equal to the maximum temperature drift rate for the signal.

[0045] Over a long period of time, the average road bank angle is considered to be zero. Any time a non-

zero value is detected, lateral acceleration offset is adjusted in the controller 26 such that lateral acceleration will drive road bank angle to zero.

[0046] Over a long period of time, the average road pitch angle is considered to be zero. Any time a non-zero value is detected, longitudinal acceleration offset is adjusted in the controller 26 such that longitudinal acceleration will drive pitch angle to zero. Road pitch angle is calculated by taking the current pitch angle estimate and then subtracting the relative pitch angle and the pitch angle offset. Logic to illustrate this includes:

```
RoadPitchEst = PitchAngleEst - PitchOffset - PitchRelative;
If (RoadPitchEst > 0) AxOffset -= MAX_AX_DRIFT_RATE;
Else if (RoadPitchEst < 0)
AxOffset += MAX_AX_DRIFT_RATE;
If (!(GET_SUSTAINED_POS_TURNING_15SEC))
{if (FltYawRate > 0)
YawRateOffset += MAX_YR_DRIFT_RATE;
If (!(GET_SUSTAINED_NEG_TURNING_15SEC))
{if (RollRate > 0)          RollRateOffset += MAX_RR_DRIFT_RATE;}
if (InstBankAngleEstimate < 5 && InstBankAngleEstimate > 0)
AyOffset += MAX_AY_DRIFT_RATE;}
if (!(GET_SUSTAINED_NEG_TURNING_15SEC))
{if (FltYawRate < 0)
YawRateOffset -= MAX_YR_DRIFT_RATE;
If (!(GET_SUSTAINED_POS_TURNING_15SEC))
{if (RollRate < 0)
```

```
    RollRateOffset -= MAX_RR_DRIFT_RATE;}
    if (InstBankAnbleEstimate > -5 &&
InstBankAngleEstimate < 0)
    AyOffset -= MAX_AY_DRIFT_RATE;
```

[0047] In operation block 66, a third offset correction signal is generated for a vehicle dynamic sensor when the vehicle 19 is at rest. Embodiments thereof are included below.

[0048] An example of offset resting external inputs includes:

RSC Code Name	**Logic Description Name**
1. bf_uc_INITIALIZATION_STATUS: Flag to determine if initialization is completed	INITIALIZATION_COMPLETE
2. ss_dps_ROLL_ACCELERATION2: Roll Acceleration	RollAcceleration
3. ss_dps_YAW_ACCELERATION2: Yaw acceleration	YawAcceleration

[0049] An example of offset resting state variables includes:

RSC Code Name	**Logic Description Name**
1. ss_dps_FLT_YR_COMP_REST: Filtered and compensated resting yaw rate	
2. ss_dps_RESTING_YR_OFFSET: Resting slow drifting offset.	RestingYROffset
3. bf_uc_OFFSET_FLAG: Flags that are set during resting	VEHICLE_STOPPED_FOR_ONE_SEC
4. ss_mpss_FLT_LONG_ACC_RESTING: Initial resting longitudinal acceleration	HOLD_RESTING_VALUES
5. ss_mpss_FLT_LAT_ACC_RESTING: Initial resting lateral acceleration	FltAyResting
6. ss_RESTING_UPDATE_CLOCK: Clock	FltAxResting

[0050] An example of offset resting outputs includes:

RSC Code Name	**Logic Description Name**
1. CLEAR_SUSTAINED_POS_TURNING _15SEC: Clear the continuous 15 second turn counter	
2. CLEAR_SUSTAINED_NEG_TURNING _15SEC: Clear the continuous 15 second turn counter	
3. ss_dps_YAW_RATE_OFFSET	YawRateOffset
4. ss_dps_ROLL_RATE_OFFSET	RollRateOffset
5. ss_dps_LONG_ACC_OFFSET	AxOffset
6. ss_dps_LAT_ACC_OFFSET	AyOffset

[0051] An example of offset resting parameters includes: 1. Yr_ofst_eep_max; 2. ALLOWED_RESTING_ROLL_ACC: Acceleration that is allowed when standing still; 3. ALLOWED_RESTING_YAW_ACC: Acceleration that is allowed when standing still.

[0052] After the vehicle 19 comes to a complete stop, to ensure that no transient signals (such as may be caused by the vehicle 19 rocking, bouncing, or swaying on its suspension) are received in the controller 26, the controller 26 waits, for example, an additional second. An example of logic thereof includes:

```
If ((INIT_FLAG & INITIALIZATION_COMPLETE)
&& (!(RSC_SENSORS_DISTURBED)))
{if ( !( OFFSET_FLAG & VEHICLE_STOPPED_FOR_ONE_SEC)
{RESTING_UPDATE_CLOCK ++;
If (RESTING_UPDATE_CLOCK >= ONE_SEC_UP)
OFFSET_FLAG|= VEHICLE_STOPPED_FOR_ONE_SEC}}
```

[0053] An average value for filtered lateral acceleration (Ax) and longitudinal acceleration (Ay) for a period of 1 second is generated after the vehicle 19 has come to a stop. These values are used to compute the Ax and Ay offsets during standing still. Any drifts in Ax and Ay are considered to be due to sensor drifts and is compensated for by applying the appropriate offset. An example of logic and inputs therefore follow.

[0054] One example of filtering and compensation external inputs includes:

RSC Code Name	**Logic Description Name**
1. Get_vehicle_standstill()	
2. bf_uc_INITIALIZATION_STATUS	
3. ss_dps_YAW_RATE	RawYawRate
4. ss_dps_ROLL_RATE	RawRollRate
5. ss_mpss_LONG_ACC	RawAx
6. ss_mpss_LAT_ACC	RawAy
7. ss_mpss_YAW_RATE_OFFSET	YawRateOffset
8. ss_mpss_ROLL_RATE_OFFSET	RollRateOffset
9. ss_mpss_LONG_ACC_OFFSET	AxOffset
10. ss_mpss_LAT_ACC_OFFSET	AyOffset
11. ss_dps_FLT_ROLL_RATE	FltRollRate
12. ss_dps_FLT_YAW_RATE	FltYawRate
13. ss_mpss_FLT_LONG_ACC	FltAx
14. ss_mpss_FLT_LAT_ACC	FltAy

[0055] One example of filtering and compensation state variables includes:

RSC Code Name	Logic Description Name
1. uc_LONG_TERM_COMP_UPDATE_CLOCK	
2. ss_dps_YR_COMP_UNFLT	RawYawRate
3. ss_dps_RR_COMP_UNFLT	RawRollRate
4. ss_dps_AX_COMP_UNFLT	RawAx
5. ss_dps_AY_COMP_UNFLT	RawAy
6. uc_RR_POS_CNTR	RR_POS_CNTR
7. uc_RR_NEG_CNTR	RR_NEG_CNTR
8. uc_YR_POS_CNTR	YR_POS_CNTR
9. uc_YR_NEG_CNTR	YR_NEG_CNTR
10. uc_AX_POS_CNTR	AX_POS_CNTR
11. uc_AX_NEG_CNTR	AX_NEG_CNTR
12. uc_AY_POS_CNTR	AY_POS_CNTR
13. uc_AY_NEG_CNTR	AY_NEG_CNTR

[0056] One example of filtering and compensation outputs includes:

RSC Code Name	Logic Description Name
1. bf_uc_OFFSET_FLAG	
2. ss_dps_FLT_ROLL_RATE	FltRollRate
3. ss_dps_FLT_YAW_RATE	FltYawRate
4. ss_mpss_FLT_LONG_ACC	FltAx
5. ss_mpss_FLT_LAT_ACC	FltAy

[0057] One example of filtering and compensation parameters includes: RR_MAX_LOOP; RR_DECR_CNTR; YR_MAX_LOOP; YR_DECR_CNTR; AX_MAX_LOOP; AX_DECR_CNTR; AY_MAX_LOOP; and AY_DECR_CNTR.

[0058] Filtering and compensation logic includes the computed values of the offsets taken out from the raw value before they are filtered. Therefore:

RawRollRate -= RollRateOffset; RawYawRate -= YawRateOffset; RawAx -= AxOffset; and RawAy -= AyOffset.

[0059] An embodiment of logic for generating offsets during standing still includes:

```
if ((OFFSET_FLAG & VEHICLE_STOPPED_FOR_ONE_SEC)
&& ! (OFFSET_FLAG & HOLD_RESTING_VALUES))
{FltAxResting = ((FltAxResting * TempCounter) + FltAx) / (TempCounter + 1);
FltAyResting = ((FltAyResting * TempCounter) + FltAy) / (TempCounter + 1);
TempCounter ++;
If (TempCounter >= ONE_SEC_UP)
OFFSET_FLAG |= HOLD_RESTING_VALUES;}
```

[0060] If any vehicle rolling motions or change in yaw is picked up when stopped, VEHICLE_STOPPED_FOR_ONE_SEC and HOLD_RESTING_VALUES flags is unset. Resting Ax and Ay values (step 2) is re-computed after this. Logic therefore includes:

```
If (INIT_FLAG & INITIALIZATION_COMPLETE)
{if (( RollAcceleration <= VEHICLE_STAND_STILL_ROLL_ACC)
&& (YawAcceleration <= VEHICLE_STAND_STILL_YAW_ACC))
{OFFSET_FLAG &= ~VEHICLE_STOPPED_FOR_ONE_SEC
OFFSET_FLAG &= ~HOLD_RESTING_VALUES;}}
```

[0061] Any changes in lateral acceleration and longitudinal acceleration are assumed to be due to sensor drifts when the vehicle 19 is at a standstill. Logic to illustrate this includes:

```
If ((VEHICLE_STOPPED_FOR_ONE_SEC)
&& (HOLD_RESTING_VALUES)
&& (ROLL_ACCELERATION < ALLOWED_RESTING_ROLL_ACC)
&& (YAW_ACCELERATION < ALLOWED_RESTING_YAW_ACC))
{if (FltAx > FltAxResting)
FltAxOffset += MAX_AX_DRIFT_RATE;
Else if (FltAx < FltAxResting)
FltAxOffset -= MAX_AX_DRIFT_RATE;
if (FltAy > FltAyResting)
FltAyOffset += MAX_AY_DRIFT_RATE;
Else if (FltAy < FltAyResting)
FltAyOffset -= MAX_AY_DRIFT_RATE;}
```

[0062] During a vehicle standstill, offset compensation is performed within the controller 26 such that the filtered roll and yaw rate is zero. Logic to illustrate this includes:

```
If (VEHICLE_STOPPED_FOR_ONE_SEC)
{Temp = RESTING_TIME_CONSTANT / LOOP_TIME;
RollRateOffset += FltRollRate * Temp;
YawRateOffset += FltYawRate * Temp;}
```

[0063] If the vehicle 19 is turned while standing still, like when the vehicle 19 is stopped/parked on a turntable of a parking facility (referred to as a turntable event), the offset compensation is held constant until the turning is completed by calculating a resting yaw rate compensation value while the vehicle 19 is standing still. No fast compensation is performed while the vehicle 19 is standing still for the resting yaw rate. If this value exceeds a threshold, offset compensation is paused until it is

below the threshold. Logic to illustrate this includes:

```
FltYRRest = FLTR_COEFF * FltYRRest + (1 - FLTR_COEFF)*(RawYawRate - RestingYROffset;
If (FltrYRRest > 0)
RestingYROffset += MAX_YR_DRIFT_RATE;
Else if (FltrYRRest < 0)
RestingYROffset -= MAX_YR_DRIFT_RATE;
If (FltrYRRest > MAX_REST_YR)
{OFFSET_FLAG &= ~ VEHICLE_STOPPED_FOR_ONE_SEC;
OFFSET_FLAG &= ~ HOLD_RESTING_VALUES;}
```

[0064] If an initialization has taken place during a turn table type event, it is corrected when the vehicle 19 is standing still after initialization. In such event, the offsets value is reset to the average of the maximum and the minimum values from EEPROM 47. In such cases, the rate sensors (27, 28, 30, or 31) are limited to, for example, +-3.5deg/sec and the acceleration sensors is limited to, for example, +-1.5m/sec^2. The controller 26 achieves this by taking the initial value of the yaw rate after the completion of offset initialization and FltYRRest. Logic to illustrate this includes:

```
If (INTIALIZATION_JUST_FINISHED)
{if (FltYRInitial > FltYRRest + 2)
|| (FltYRInitial < FltYRRest - 2)
{RollRateOffset = LIMIT(+-3.5,(MAX_RR_OFST_EE + MIN_RR_OFST_EE )/2);
YawRateOffset = LIMIT(+-3.5,(MAX_YR_OFST_EE + MIN_YR_OFST_EE)/2);
AxOffset = LIMIT(+-1.5,(MAX_AX_OFST_EE + MIN_AX_OFST_EE )/2);
```

AyOffset = LIMIT(+-1.5, (MAX_AY_OFST_EE + MIN_AY_OFST_EE)/2);}}

[0065] In operation block 68, the vehicle dynamic sensor (27, 28, 30, or 31) is corrected in response to the first offset correction signal, the second offset correction signal, and the third offset correction signal.

[0066] In other words, the computed values of the offsets are taken out from the raw sensor value before the sensors (27, 28, 30, or 31) are filtered. Logic to illustrate this includes:

```
RawRollRate -= RollRateOffset;
RawYawRate -= YawRateOffset;
RawAx -= AxOffset;
RawAy -= AyOffset;
```

[0067] The roll rate sensor 31 is filtered by limiting the roll rate velocity to a maximum of RR_xxx_CNTR * RR_DELTA. These values are established by taking the worst-case conditions (the maximum roll rate) from data. The slope is continually increased by increasing the counter. The counter is limited the RR_MAX_LOOP value. So, the maximum rate change experienced by the roll rate sensor 31 is RR_MAX_LOOP * RR_DELTA. Logic to illustrate this includes:

```
if (RawRollRate > FltRollRate)
{if (RR_POS_CNTR < RR_MAX_LOOP) RR_POS_CNTR ++;
if (RR_NEG_CNTR >= RR_DECR_CNTR) RR_NEG_CNTR -= RR_DECR_CNTR;
else RR_NEG_CNTR = 0;
FltRollRate += RR_POS_CNTR * RR_DELTA;
```

If (RawRollRate < FltRollRate) FltRollRate = RawRollRate;}

else {if (RR_NEG_CNTR < RR_MAX_LOOP) RR_NEG_CNTR ++;

if (RR_POS_CNTR >= RR_DECR_CNTR) RR_POS_CNTR -= RR_DECR_CNTR;

else RR_POS_CNTR = 0;

FltRollRate -= RR_NEG_CNTR * RR_DELTA;

If (RawRollRate > FltRollRate) FltRollRate = RawRollRate;}

[0068] The same logic is applied to filter Yaw Rate:

```
if (RawYawRate > FltYawRate)
{if (YR_POS_CNTR < YR_MAX_LOOP) YR_POS_CNTR ++;
if (YR_NEG_CNTR >= YR_DECR_CNTR) YR_NEG_CNTR -= YR_DECR_CNTR;
else YR_NEG_CNTR = 0;
FltYawRate += YR_POS_CNTR * YR_DELTA;
If (RawYawRate < FltYawRate) FltYawRate = RawYawRate;}
else{if (YR_NEG_CNTR < YR_MAX_LOOP) YR_NEG_CNTR ++;
if (YR_POS_CNTR >= YR_DECR_CNTR) YR_POS_CNTR -= YR_DECR_CNTR;
else YR_POS_CNTR = 0;
FltYawRate -= YR_NEG_CNTR * YR_DELTA;
If (RawYawRate > FltYawRate) FltYawRate = RawYawRate;}
```

[0069] The same logic is applied to filter Longitudinal Acceleration:

```
if (RawAx > FltAx)
```

```
{if (AX_POS_CNTR < AX_MAX_LOOP) AX_POS_CNTR ++;
if (AX_NEG_CNTR >= AX_DECR_CNTR) AX_NEG_CNTR -= AX_DECR_CNTR;
else AX_NEG_CNTR = 0;
FltAx += AX_POS_CNTR * AX_DELTA;
If (RawAx < FltAx) FltAx = RawAx;}
else {if (AX_NEG_CNTR < AX_MAX_LOOP) AX_NEG_CNTR ++;
if (AX_POS_CNTR >= AX_DECR_CNTR) AX_POS_CNTR -= AX_DECR_CNTR;
else AX_POS_CNTR = 0;
FltAx -= AX_NEG_CNTR * AX_DELTA;
If (RawAx > FltAx) FltAx = RawAx;}
```

[0070] The same logic is applied to filter Lateral Acceleration:

```
if (RawAy > FltAy)
{if (AY_POS_CNTR < AY_MAX_LOOP) AY_POS_CNTR ++;
if (AY_NEG_CNTR >= AY_DECR_CNTR) AY_NEG_CNTR -= AY_DECR_CNTR;
else AY_NEG_CNTR = 0;
FltAy += AY_POS_CNTR * AY_DELTA;
If (RawAy < FltAy) FltAy = RawAy;}
Else
{if (AY_NEG_CNTR < AY_MAX_LOOP) AY_NEG_CNTR ++;
if (AY_POS_CNTR >= AY_DECR_CNTR) AY_POS_CNTR -= AY_DECR_CNTR;
else AY_POS_CNTR = 0;
FltAy -= AY_NEG_CNTR * AY_DELTA;
If (RawAy > FltAy) FltAy = RawAy;}
```

[0071] While particular embodiments of the invention have been shown and described, numerous

variations and alternate embodiments will occur to those skilled in the art. Accordingly, it is intended that the invention be limited only in terms of the appended claims.